LA-UR-15-27726
Approved for public release; distribution is unlimited.

Title: Final Presentation

Author(s): Dutta, Soumya
Canada, Curtis Vincent

Intended for: Web

Issued: 2015-10-05

Final Presentation

Soumya Dutta
Summer Student, Data Science at Scale group, LANL
Ph.D. Student, The Ohio State University

July 23, 2015

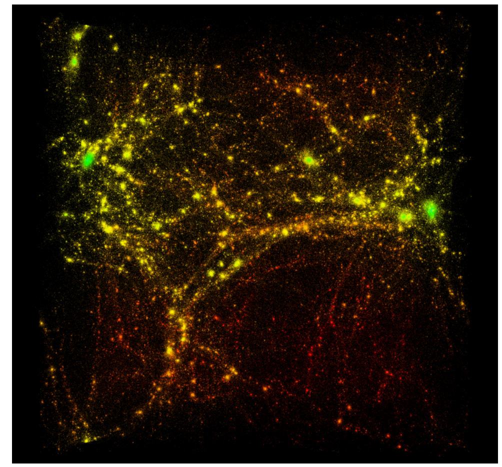
Topics to be covered:

1. Discussion on several data representations and a global algorithm comparison framework
   – Why it is needed?
   – How it can be done efficiently?
   – A framework for comparison among the data representations
2. In-Situ early Convergence detection on a Monte Carlo based simulation called openMC

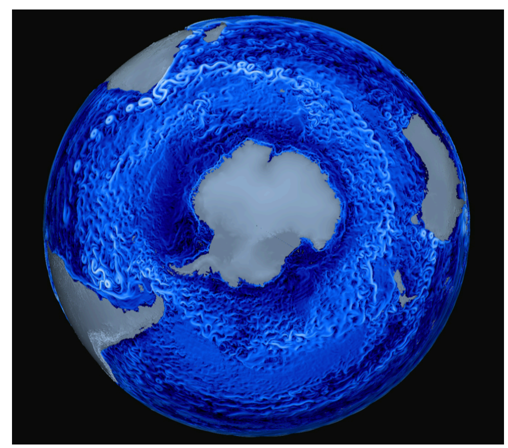
Various Data summarization techniques and a framework to compare them

Efficient data representations

- Impossible to store all the raw data
  - Large size (Petabyte ~ Exabyte)
  - Bottleneck in I/O
  - Flops are free, not the disk space
- Efficient data summarization techniques are needed
  - Reduce the size of the data
  - Still preserve necessary details
  - Answer domain specific questions



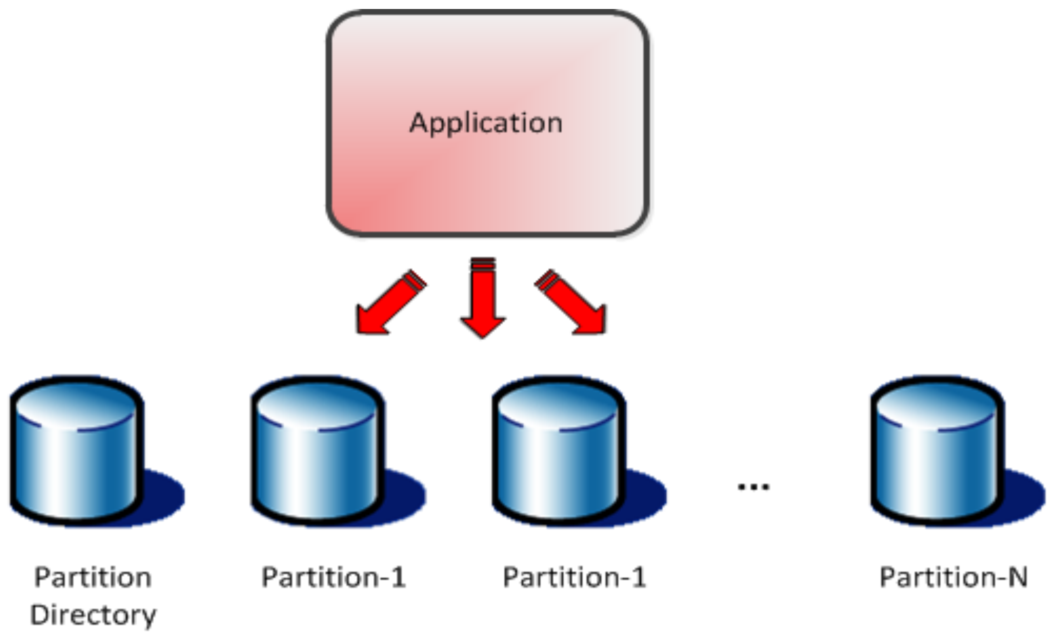
Particles in cosmology data



MPAS ocean simulation

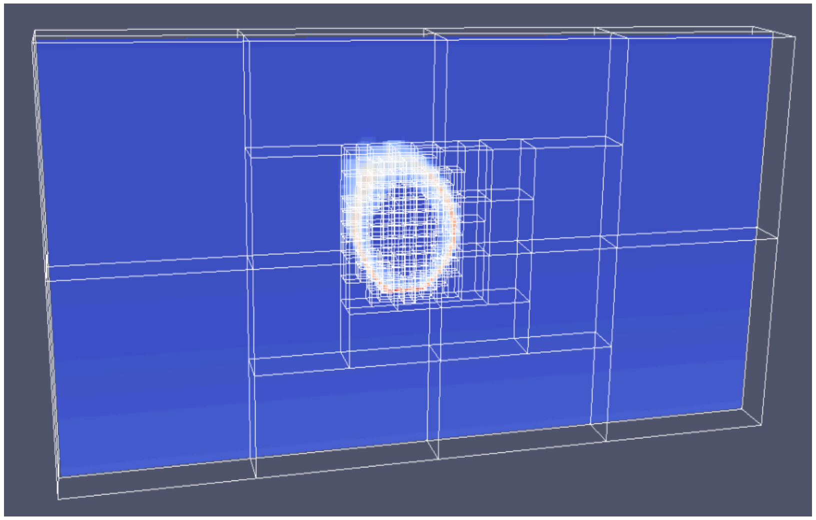
Create Data Representations

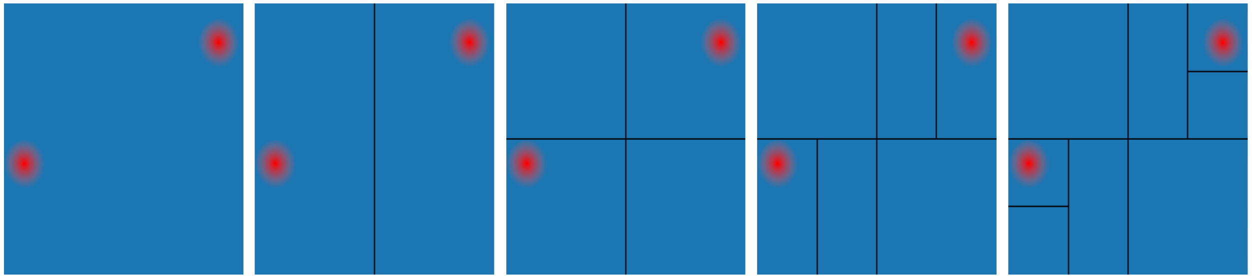
- Prioritization of data
  - An In-Situ framework
  - Partitioning and Summarizing
  - Estimation of error in the data representation scheme



Data Representations

- Partitioning Schemes:
  - Kd-tree based partitioning
  - Voronoi tessellation
  - Distributions (In future)





An illustrative partitioning example

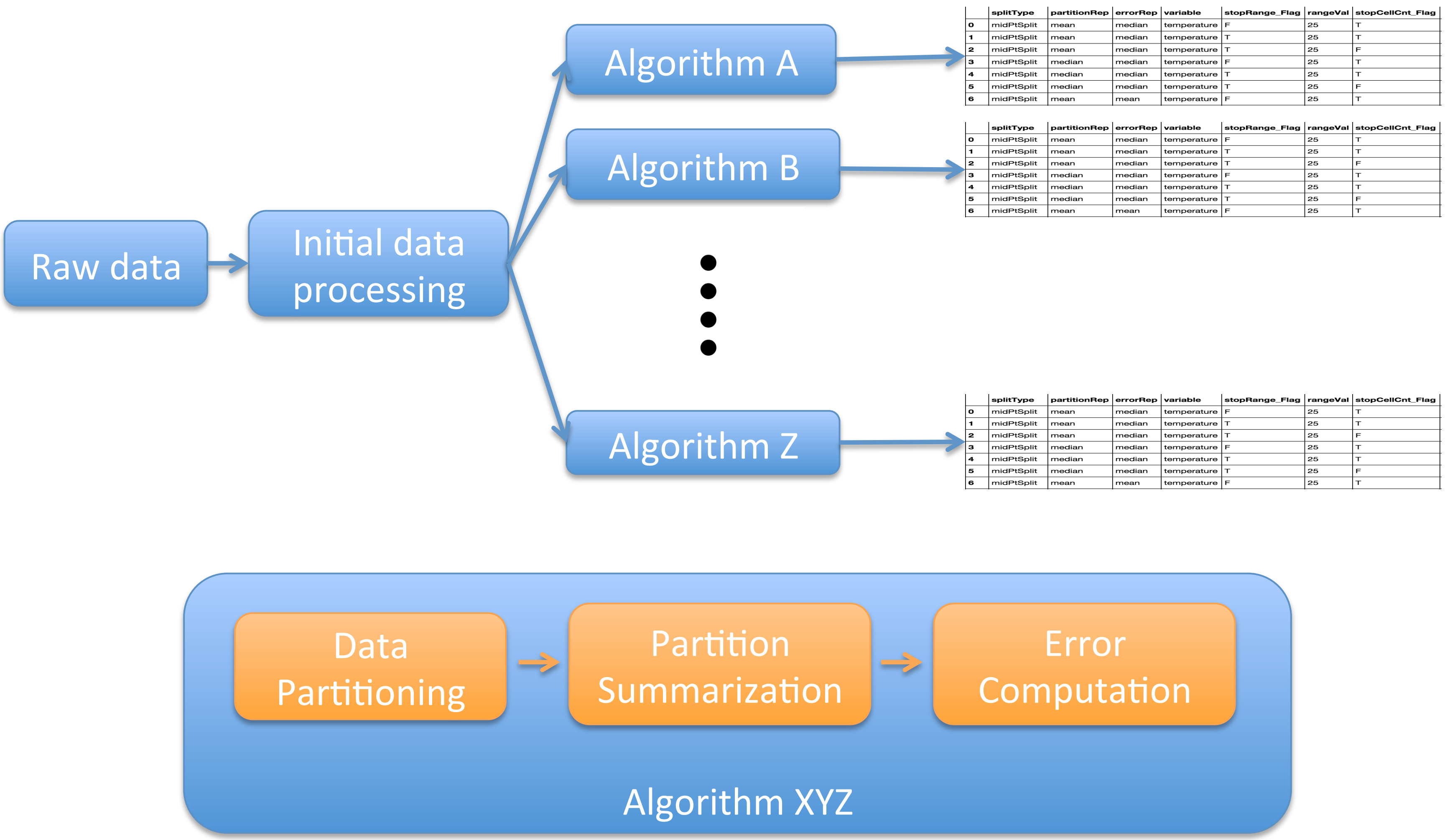
Partition Summarization

- Find a representation for the partitions
  - Mean, Median or Midpoint
- Estimate the quality of the partitioning and the incurred error
  - Sum of squared error (sse)
  - pAIC

A generalized framework for comparing across data representations

- In order to compare across different schemes we need a comparison framework
- A python based Score-boarding framework
- Goals:
  - A global scale parameter study on the parameter space of the representations
    - Storage requirements for the representation
    - Error of the representation

A generalized framework for comparing across data representations



A generalized framework for comparing across data representations

- Run on all parameter combinations
- Final product is a database table
  - Keeps track of all the parameters used for a run
  - Can be queried efficiently to order based on different parameters
  - Each representation will have their own parameter study table
  - Multiple tables can be joined and compared for finding the best parameter combinations

Some test parameters and results

- Comparing data partitioning schemes:
  - Kd-tree partitioning
  - Voronoi tessellation
  - Distributions (In future)
- Partition representations
  - Mean
  - Median
  - Midpoint
- Dimensions used to split
- Stopping Criteria
  - Max entropy of a partition
  - Specific value range of a variable
  - Max tree depth
- Error Metric
  - pAIC

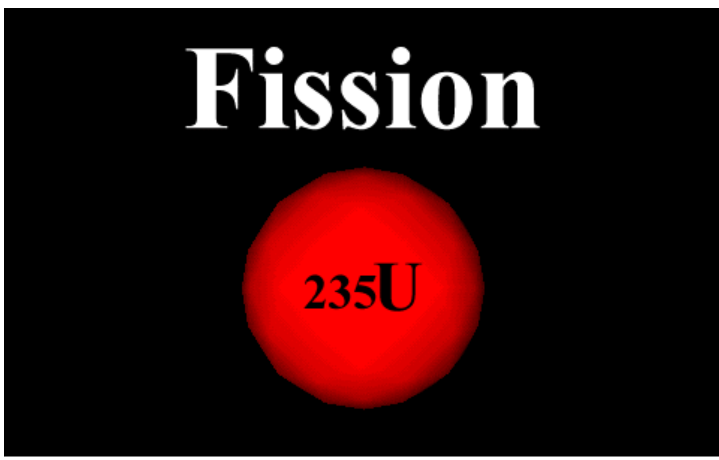
Summarization Results ordered by pAIC

	partitionRep	errorRep	use_randomized_init_points	min_cell_area	refinement_metric	max_depth	storage	raw_Size	variable	pAIC	sse	estErr
0	median	max	F	24	Average_abs_error	50	0.01002	2.2102	TEMP	0.00150	136.54635	1198.00510
1	median	max	F	8	Average_abs_error	50	0.01012	2.2102	TEMP	0.00151	138.34750	1194.39175
2	median	max	F	16	Average_abs_error	50	0.01008	2.2102	TEMP	0.00151	139.01264	1205.51937
3	median	max	T	8	Average_abs_error	50	0.01428	2.2102	TEMP	0.00172	145.36286	1297.65176
4	median	max	T	16	Average_abs_error	50	0.02208	2.2102	TEMP	0.00230	189.37300	1566.22197
5	median	max	T	24	Average_abs_error	50	0.02792	2.2102	TEMP	0.00253	193.12431	1609.63746
6	mean	max	F	16	Average_abs_error	50	0.07873	2.2102	TEMP	0.00447	264.13807	1910.36650
7	mean	max	F	24	Average_abs_error	50	0.07873	2.2102	TEMP	0.00447	264.13807	1910.36650
8	mean	max	F	8	Average_abs_error	50	0.07879	2.2102	TEMP	0.00448	263.87691	1909.07034
9	mean	max	T	8	Average_abs_error	50	0.08888	2.2102	TEMP	0.00484	276.08041	1974.23433
10	mean	max	T	16	Average_abs_error	50	0.09620	2.2102	TEMP	0.00521	288.23234	2085.16615
11	mean	max	T	24	Average_abs_error	50	0.10517	2.2102	TEMP	0.00554	298.03379	2143.74996
12	midpt	max	F	24	Average_abs_error	50	0.15779	2.2102	TEMP	0.00887	667.00535	3692.38774
13	midpt	max	F	8	Average_abs_error	50	0.15786	2.2102	TEMP	0.00888	665.16715	3699.67178
14	midpt	max	F	16	Average_abs_error	50	0.15786	2.2102	TEMP	0.00888	665.16715	3699.67178
15	midpt	max	T	8	Average_abs_error	50	0.16768	2.2102	TEMP	0.00921	668.97354	3762.93955
16	midpt	max	T	16	Average_abs_error	50	0.17469	2.2102	TEMP	0.00947	673.65780	3790.43463
17	midpt	max	T	24	Average_abs_error	50	0.18449	2.2102	TEMP	0.00988	681.35475	3931.93283
18	median	mean	F	24	Average_abs_error	50	0.02792	2.2102	TEMP	0.12227	193.12431	134285.33726
19	median	mean	F	16	Average_abs_error	50	0.02799	2.2102	TEMP	0.12270	190.87622	134757.53595
20	median	mean	F	8	Average_abs_error	50	0.02800	2.2102	TEMP	0.12284	189.43835	134905.42032
21	median	mean	T	8	Average_abs_error	50	0.03637	2.2102	TEMP	0.15494	207.57115	170178.78300
22	median	mean	T	16	Average_abs_error	50	0.04422	2.2102	TEMP	0.19174	224.96519	210659.28119
23	median	mean	T	24	Average_abs_error	50	0.05348	2.2102	TEMP	0.23212	246.79074	255054.14733
24	median	median	F	24	Average_abs_error	50	0.05348	2.2102	TEMP	0.23212	246.79074	255054.14733
25	median	median	F	8	Average_abs_error	50	0.05355	2.2102	TEMP	0.23229	247.43829	255241.46211

In-Situ early Convergence detection in openMC

openMC: Monte Carlo Particle transport code

- OpenMC simulates neutron moving around randomly in a nuclear reactor



Goal of the work

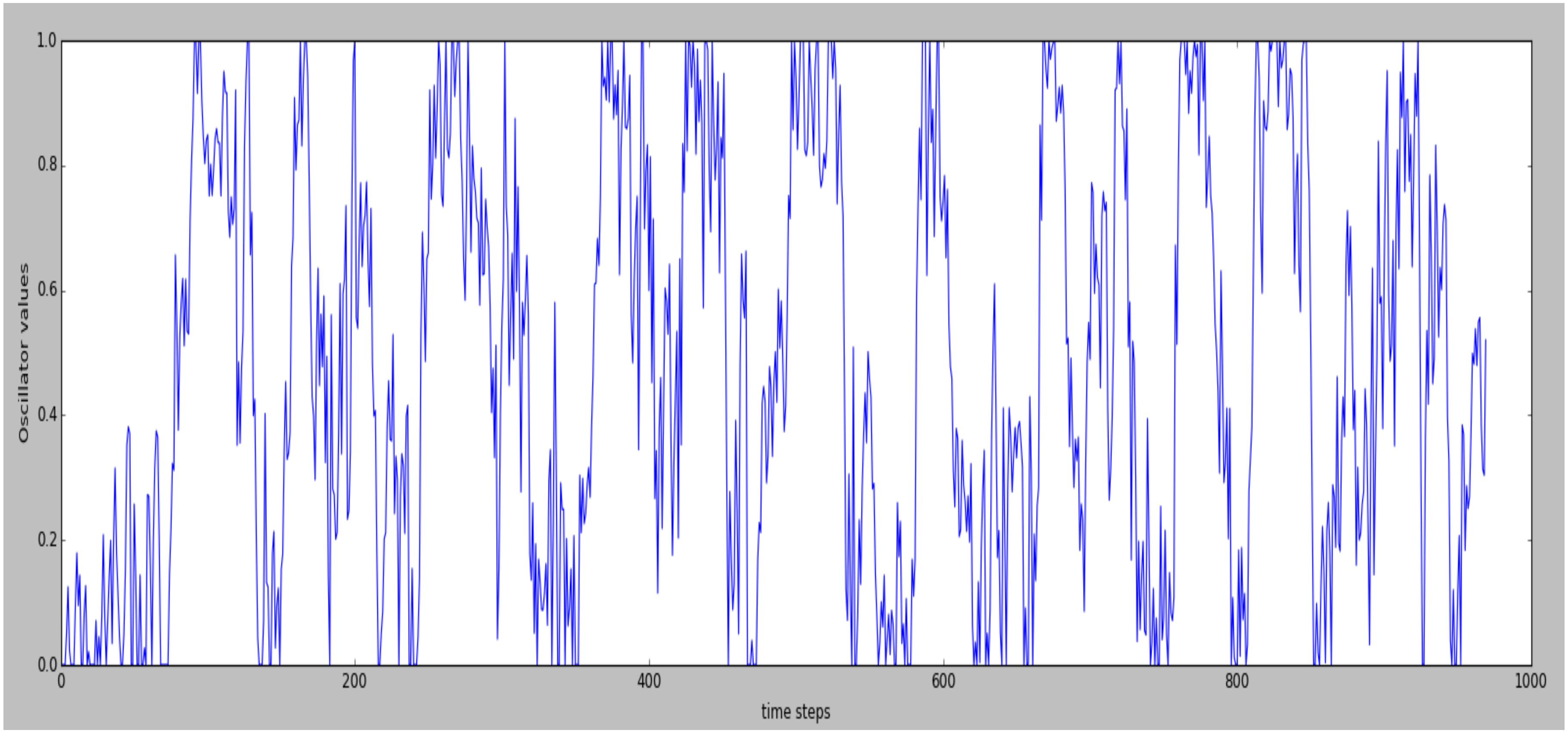
- Run the simulation code
- Develop a Monte Carlo simulation convergence test
- Inject the convergence test code into simulation
- Test for early convergence detection
- Conduct a scale study for performance estimation

Stochastic Oscillator in early Convergence Detection

- Convergence is detected using the Entropy values of source distributions
- The stationarity of Entropy values reflect the convergence
- When convergence is reached:
  - The expected value of the stochastic oscillator will be 0.5

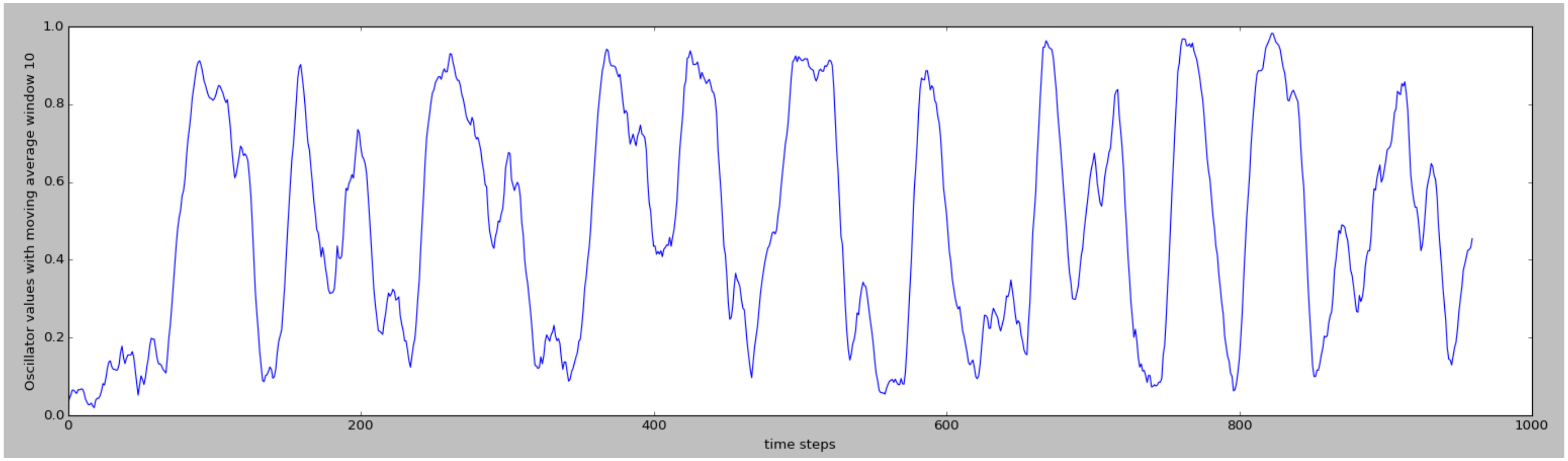
- Ref: Application of the stochastic oscillator to assess source convergence in monte carlo criticality calculations, Paul K. Romano, M&C 2009.

Results obtained with Stochastic Oscillator

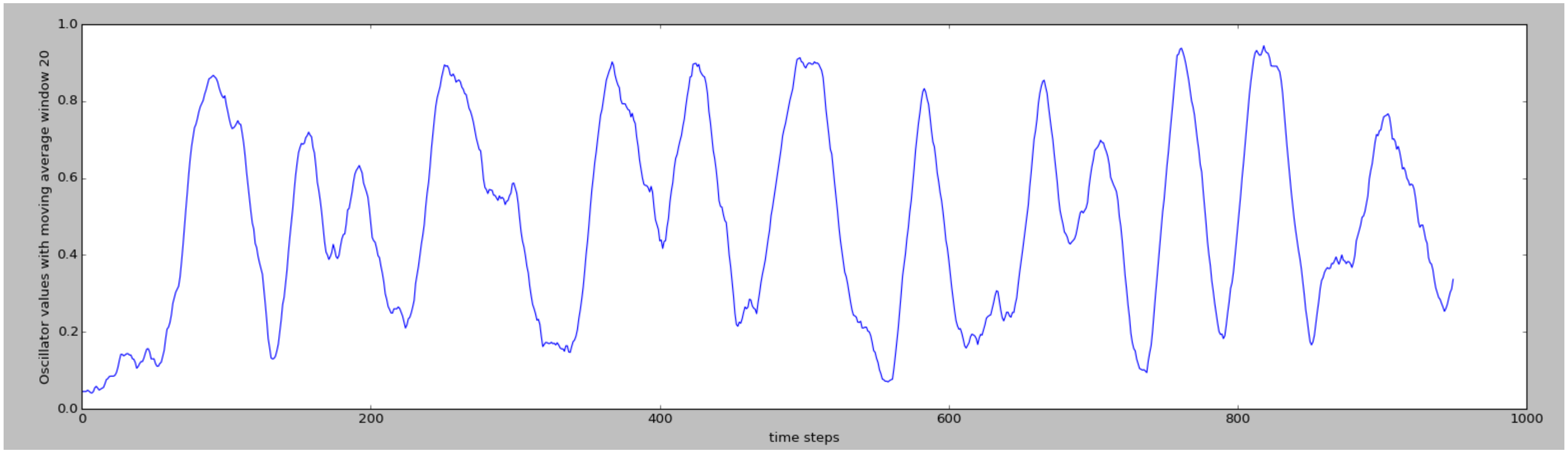


Result of the Stochastic Oscillator with a window of size 30

Results obtained with Stochastic Oscillator

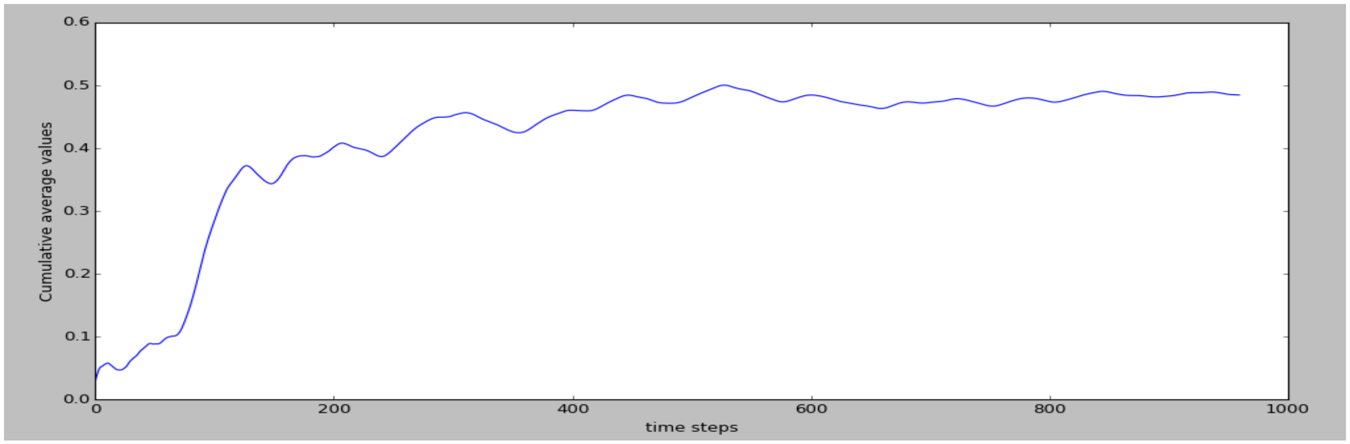


Result of the Stochastic Oscillator with a smoothing window of size 10

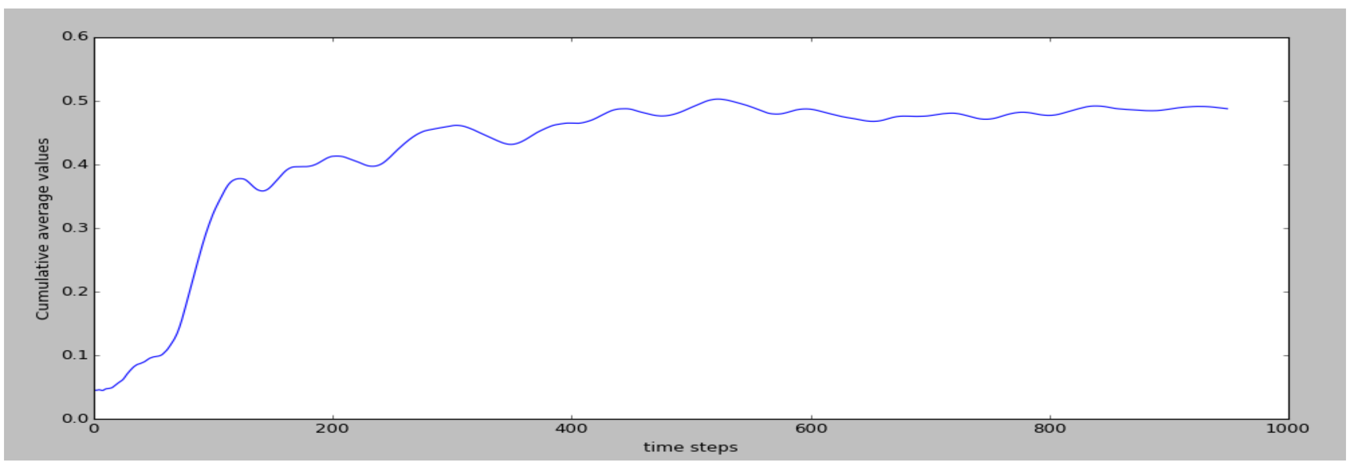


Result of the Stochastic Oscillator with a smoothing window of size 20

Results obtained with Stochastic Oscillator



Cumulative average of the values of Oscillator with a smoothing window of size 10



Cumulative average of the values of Oscillator with a smoothing window of size 20

Some other notes

- I wrote a converter from VTK multi-block unstructured dataset to SQLite3 database.
- Another converter from VTI to SQLite3 database.

- Got familiar with R
- Got used to Mac!

Thank You!